Github README

TapData: Revolutionizing Enterprise Data Integration with Real-Time & Caching

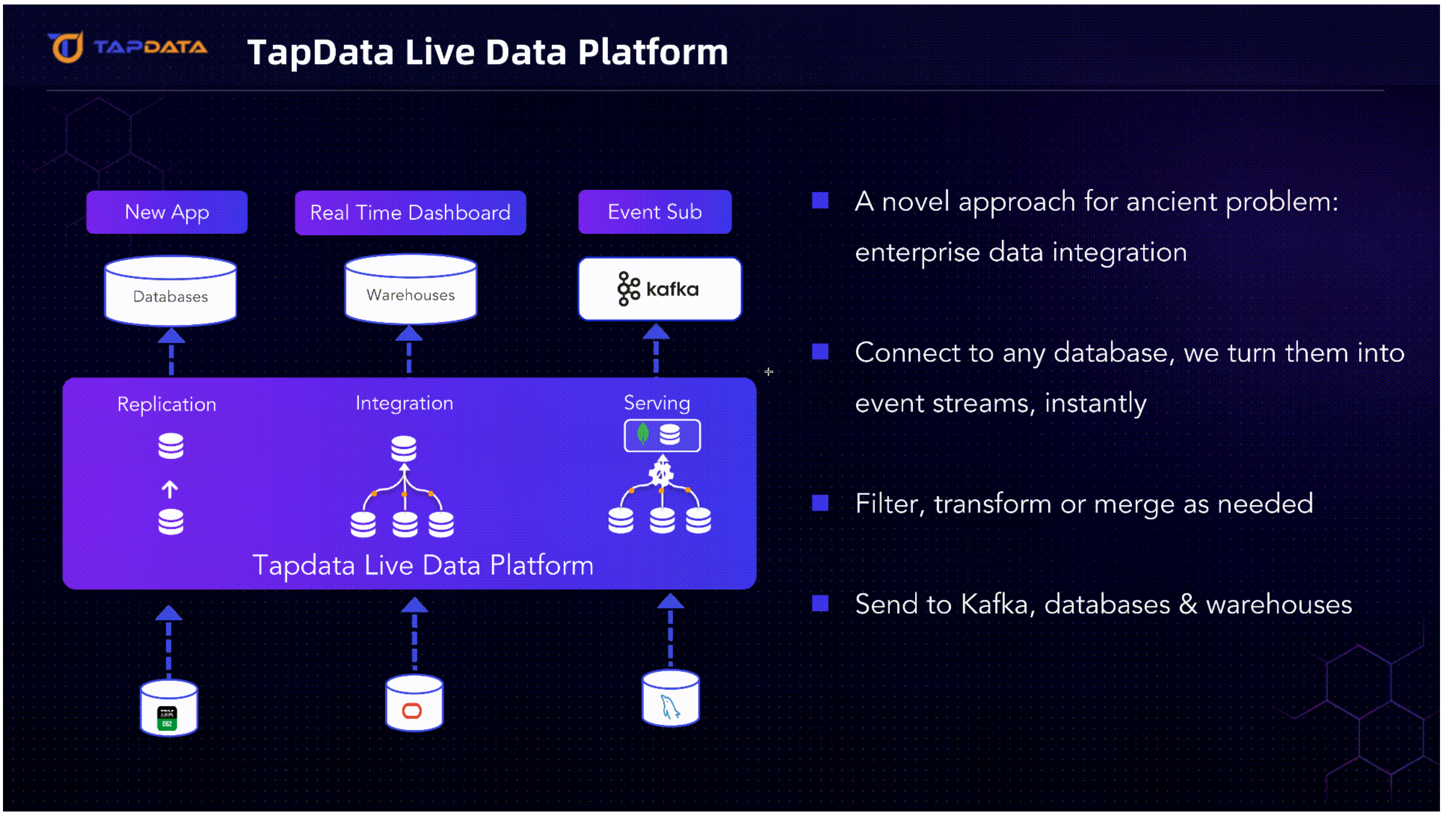
Overview

TapData is an open source, real-time data platform designed to solve the age-old data integration problem with a novel approach:

- Uses CDC-based, real-time data pipelines instead of batch-based ETL
- Supports a centralized data hub architecture, in addition to point-to-point

We believe real-time data brings an unmatched user experience and is even more applicable for future LLM applications. We also believe that providing a hub architecture can significantly reduce the number of pipelines needed for serving multiple applications.

How It Works



- A novel approach for ancient problem: enterprise data integration
- Connect to any database, we turn them into event streams, instantly
- Filter, transform or merge as needed
- Send to Kafka, databases & warehouses

- First, connect to your existing application databases using built-in connectors. You need to prepare your network access and credentials to the databases, and some configuration of your databases may be required.
- TapData will monitor your database's log files(redo log, binlog etc) and capture the changes (inserts/updates/deletes)
- TapData will turn the change events into an event stream with the full record as the payload
- You can then send the record to Kafka, another database or a data warehouse
- You can also store the data in the TapData platform to serve data queries from your applications using APIs

Primary Use Cases

You may use TapData for use cases where low latency data transfer requirement is paramount. Examples include:

- Feeding data into data warehouse
- Synchronizing your data from RDBMS to MongoDB, Redis, Elastic for query acceleration
- Generate event streams for Kafka
- Heterogenius database replication
- Building a centralized data platform

Quick comparison with alternatives

Compare To	TapData's Differentiation
Fivetran	• Fivetran is best for SaaS sources, for databases it is not cost-effective and does not support real-time in the cloud • TapData specializes in databases and real-time transfer
Airbyte	• Airbyte is mostly designed for batch ETL, it only has a handful CDC connectors while TapData has complete coverage for most TP databases https://docs.airbyte.com/understanding-airbyte/cdc/ • Airbyte only supports point to point, while TapData supports a Hub architecture
Informatica / Kettle	• Both Informatica and Kettle are designed for batch ETL processing • TapData is designed for real-time processing
Kafka Data Pipelines	

	• For Kafka, the user is responsible to send data to Kafka as well as write code to consume it • TapData is a no code/low code solution, it only needs to be configured You may read an indepth analysis about between Kafka ETL vs. TapData.
Oracle Golden Gate / Attunity	• Both are real-time database replication tools similar to TapData • TapData provides more deployment options: cloud and community • TapData supports multi-table merging, materialized view • TapData supports hub architecture
FlinkCDC	• Requires Flink dependency •
Debezium	

Features

- 60+ Built-in CDC connectors
- Point to point or hub architecture
- No code with drag & drop experience
- Javascript & Python UDF (user defined functions) in the pipeline
- Centralized caching: Reduces the need for numerous data pipelines, optimizing performance and resource utilization
- Heterogeneous database support, from SQL to NoSQL or vice versus
- Multi-table join / Building materialized views (beta)
- Community-driven: open-source and built with the community for the community, ensuring a solution that meets real-world needs

Getting Started

Preparation

Prepare a Linux or MacOS computer with Docker installed.

Deployement

Docker

All in one docker

```
docker run -it -p 3000:3000 ghcr.io/tapdata/tapdata:latest
```

- Using -p to specify the Docker mapping access port, for example: accessing through port 8080: -p 8080:3000.

Docker without mongodb

- When starting the Docker command, add the environment variable MONGO_URI to specify the address of MongoDB

```
docker run -it -p 3000:3000  -e MONGO_URI=mongodb://xxx.xxx.xxx.xxx:<PORT>/tapdata  ghcr.io/tapdata/tapdata:latest
```

Startup succeeded

```
<<< Start Server [SUCCESS]
All Done, Please Visit http://localhost:3000
```

Linux

Software and Hardware Requirements

- CPU：8 核
- Memory：16 GB
- Disk：100 GB
- Operating System：CentOS 7 + 或 Ubuntu 16.04 +
- JRE: Java Runtime Environment 1.8.xxx

Install MongoDB

Tapdata requires MongoDB database as a dependency for running

Install Tapdata

- Get the download link for the latest community version from GitHub releases

```
https://github.com/tapdata/tapdata/releases
```

- Extract the installation package

```
tar xvf tapdata-v3.xx.xx-<xxxxx>.tar.gz
```

- Start tapdata
  - MONGO_URI: The MongoDB database connection address, For example, the connection string may look like: `mongodb://admin:password@127.0.0.1:27017/mydb?replicaSet=xxx&authSource=admin`.

```
export MONGO_URI=<mongodb://admin:password@127.0.0.1:27017/mydb?replicaSet=xxx&authSource=admin>
./start.sh
```

- The following is the startup prompt. "All Done, Please Visit http://localhost:3000" indicates that the startup is complete.

```
  _______       _____  _____        _______
 |__   __|/\   |  __ \|  __ \   /\|__   __|/\
    | |  /  \  | |__) | |  | | /  \  | |  /  \
    | | / /\ \ |  ___/| |  | |/ /\ \ | | / /\ \
    | |/ ____ \| |    | |__| / ____ \| |/ ____ \
    |_/_/    \_\_|    |_____/_/    \_\_/_/    \_\
>>> Get Env Settings [START]
MONGO_URI  :   mongodb://root:Gotapd8!@192.168.1.184:27017/tapdata-jackin?authSource=admin
ACCESS_CODE:   3324cfdf-7d3e-4792-bd32-571638d4562f
<<< Get Env Settings [SUCCESS]
>>> Unzip Connectors [START]
<<< Unzip Connectors [SUCCESS]
>>> Start Mongo [START]
<<< Mongodb is Already Running
>>> Start Server [START]
~ Start Manager Server
Maybe tm.jar is running, please check it...
~ Start Manager Server Success
~ Waiting for Manager Server Start
* Wait Starting, Left 300 / 300 Seconds...
~ Manager server started
~ Waiting for Manager Server Start Success
~ Register all connectors
```

```
* Register Connector:
* Starting to register data sources, plan to skip data sources that are not within the registration scope
* The types of data sources that need to be registered are: [GA]
Register connector to: http://localhost:3000
* Register Connector: postgres-connector-v1.0-SNAPSHOT.jar  Starting => uploading
* Register Connector: postgres-connector-v1.0-SNAPSHOT.jar | (GA) Completed
* Register Connector: quickapi-connector-v1.0-SNAPSHOT.jar  Starting... Skipped with (Alpha)
* Register Connector: clickhouse-connector-v1.0-SNAPSHOT.jar  Starting => uploading
* Register Connector: clickhouse-connector-v1.0-SNAPSHOT.jar | (GA) Completed
* Register Connector:  Success
~ Register all connectors Success
~ Start Iengine Server
~ Start Iengine Server Success
<<< Start Server [SUCCESS]
All Done, Please Visit http://localhost:3000
```

~~Try our cloud version: https://cloud.tapdata.io~~

Documentation

Understands TapData's Concept & Architecture

Quick Tutorial:

- Oracle to Mongodb replication (https://www.youtube.com/watch?v=cDYYjSKcqKU)
- Sybase to PostgreSQL
- Building a materialized view (https://www.youtube.com/watch?v=gcJew9u2uxY&t=1s)

See complete documentation here

Contributing

We welcome contributions from the community! Whether you're interested in fixing bugs, adding new features, or improving documentation, your help is invaluable. Check out our Contributing Guidelines for more information on how to get started.

Support

If you need help or have any questions, please join our Community Forum or Slack Channel. For bug reports and feature requests, please use the Jira Issues page.

License

TapData is released under the Apache 2.0 License